

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ОДЕСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ ЗВ'ЯЗКУ ім. О.С. ПОПОВА**

Кафедра «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Ректор ОНАЗ ім. О.С. Попова

_____ **П.П. Воробієнко**

“ _____ ” _____ 2020 р.

Моделювання та верифікація програмного забезпечення

ПРОГРАМА
навчальної дисципліни
підготовки *магістра*

Спеціальність: 121 Інженерія програмного забезпечення

Одеса 2020

РОЗРОБЛЕНО ТА ВНЕСЕНО:

ОДЕСЬКОЮ НАЦІОНАЛЬНОЮ АКАДЕМІЄЮ ЗВ'ЯЗКУ ІМ. О.С.ПОПОВА

РОЗРОБНИК ПРОГРАМИ: *Шмельова Т.Р., професор кафедри КН, д.т.н., доцент*

Програму розглянуто і схвалено на засіданні кафедри КН

Протокол № ____ від “ ____ ” _____ 2020 р.

Зав. каф. _____ С.М. Вороной

Програму погоджено з кафедрами:

ІТ

Зав. каф. _____ В.В. Романюк

МЗ

Зав. каф. _____ С.А. Нікітюк

Програму розглянуто і схвалено Вченою радою ННІ ІКПІ

Спеціальність: 121 Інженерія програмного забезпечення

Протокол № ____ від “ ____ ” _____ 2020 р.

Директор ННІ ІКПІ _____ І.В. Стрелковська

Програму розглянуто і схвалено Вченою радою Одеської національної академії зв'язку ім. О.С. Попова

Протокол № ____ від “ ____ ” _____ 2020 р.

Голова ради, професор _____ О.В. Бондаренко

ВСТУП

Програма вивчення навчальної дисципліни “Моделювання та верифікація програмного забезпечення” складена відповідно до освітньо-професійної програми підготовки магістрів спеціальності *121 Інженерія програмного забезпечення*.

У цьому курсі подано баланс теорії та практики моделювання й верифікації програмного забезпечення, зокрема паралельного та розподіленого. Викладені методи являють собою основу для розробки надійного програмного забезпечення. Більшість відомостей про надійність сучасних програм ґрунтується на методах тестування, які гарантують певну вірогідність виконання програмою заданої функції. Формальне доведення коректності програм є наступним кроком підвищення надійності програмного забезпечення для спеціальних застосувань у системах реального часу, а також життєво критичних сферах.

Програма навчальної дисципліни складається з таких змістових модулів:

1. Інструментальні засоби верифікації послідовних та паралельних програм алгоритмічної мовою.
2. Моделювання та верифікація паралельного програмного забезпечення сітями Петрі.
3. Алгебра та числення процесів як засіб моделювання та верифікації розподілених програм.

1. Мета та завдання навчальної дисципліни

Метою викладання навчального курсу є формування знань основних термінів і понять математичного апарату моделювання та верифікація програмного забезпечення, вивчення теоретичних та практичних основ моделювання програмного забезпечення, принципів та основних методів верифікації програмного забезпечення, набуття практичних навичок доведення коректності прикладних алгоритмів, набуття умінь, необхідних для подальшої наукової та професійної діяльності.

Цілі курсу:

- оволодіння основними поняттями формальних засобів доведення коректності програм, зокрема відповідності програм їхнім специфікаціям, поданих у вигляді додаткових анотацій до тексту програми;
- набуття фахових навичок формалізації комп’ютерних програм у вигляді сіток Петрі та алгебри процесів, дослідження їхніх властивостей доведення коректності програм;
- набуття практичних навичок побудови основних моделей послідовних, паралельних та розподілених програм;
- ознайомлення з сучасними програмними засобами верифікації послідовних, паралельних та розподілених програм з використання методів теорії сітей Петрі та алгебри процесі ;
- набуття практичних навичок моделювання та верифікації програм сітями Петрі та алгеброю процесів.

Курс передбачає теоретичні та практичні заняття.

В межах теоретичної частини забезпечуються **знання**:

- принципи подання анотації функцій, інваріантів даних та циклів програми;

- принципи розробки моделей паралельних програм у вигляді сітей Петрі;
- методи аналізу поведінкових та структурних властивостей, редукції та декомпозиції моделей у вигляді сітей Петрі;
- принципи розробки моделей розподілених програм у вигляді алгебри процесів;
- методи перетворення моделей у вигляді алгебри процесів в системи переходів з позначками, лінеаризації та розв'язання систем буле віх рівнянь для доведення властивостей моделі;
- принципи подання властивостей моделей за допомогою темпоральної та модальної логік для перевірки моделей.

Проходження практичної частини курсу формує **вміння**:

- статичної верифікації послідовних програм з використанням інструментальних систем;
- складати анотації функцій, інваріантів та циклів за вербальними специфікаціями програм;
- розробляти моделі паралельних програм у вигляді сітей Петрі;
- застосовувати інструментальні засоби аналізу властивостей сітей Петрі для верифікації паралельних програм;
- розробляти моделі розподілених програм в формі алгебри процесів;
- застосовувати інструментальні засоби для формального доведення заданих властивостей розподілених програм;
- розробляти формальний опис властивостей програм засобами темпоральної та модальної логік за вимогами до програм вербальних специфікацій
- вміння використовувати графічні засоби подання просторів станів моделі для швидкої оцінки властивостей програм;
- вміння застосовувати комплекс засобів специфікації та верифікації ітеративно з метою вдосконалення програм відповідно до вимог коректності та ефективності функціонування.

Вивчення навчальної дисципліни “Модельовання та верифікація програмного забезпечення” передбачає формування та розвиток у студентів **компетентностей**:

загальних:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність спілкуватися іноземною мовою як усно, так і письмово;
- здатність проведення теоретичних та прикладних досліджень на відповідному рівні;
- здатність мотивувати людей та рухатися до спільної мети, працювати в команді співробітників;
- здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами з інших галузей знань/видів економічної діяльності);
- здатність удосконалювати свої навички на основі аналізу попереднього досвіду;
- здатність генерувати нові ідеї (креативність);
- здатність управляти своїм часом та розуміти важливість дедлайнів.

фахових:

- здатність аналізувати предметні галузі, формувати, аналізувати та моделювати вимоги до програмного забезпечення;
- здатність ідентифікувати, класифікувати та описувати проєктні завдання, знаходити раціональні методи й підходи до їх розв’язання;
- здатність проєктувати програмне забезпечення, включаючи проведення моделювання його архітектури, поведінки та процесів функціонування окремих підсистем і модулів;
- здатність розвивати й реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення;
- здатність оцінювати ступінь обґрунтованості застосування специфікацій, стандартів, правил і рекомендацій в професійній галузі та дотримуватися їх при реалізації процесів життєвого циклу програмного забезпечення;
- здатність ефективно керувати фінансовими, людськими, технічними та іншими проєктними ресурсами;
- здатність систематизувати професійні знання щодо створення й супроводження програмного забезпечення;
- здатність розробляти й координувати процеси, фази та ітерації життєвого циклу програмних систем на основі застосування відповідних моделей, методів та технологій розробки програмного забезпечення;
- здатність застосовувати й розвивати фундаментальні і міждисциплінарні знання для успішного розв’язання завдань інженерії програмного забезпечення з використанням та ризику знань теоретичних і практичних основ методології системного аналізу, методів формалізації системних завдань, що мають суперечливі цілі, невизначеності;
- здатність використовувати знання сучасних комп’ютерних інформаційних технологій та інструментів інженерних і наукових досліджень, розрахунків, обробки та аналізу даних, моделювання та оптимізації;
- здатність демонструвати та застосовувати на практиці знання методів моделювання програмних систем, оцінювання їхньої ефективності та якості.

Результати навчання даної дисципліни деталізують такі **програмні результати навчання**:

- знати й системно застосовувати методи аналізу та моделювання прикладної галузі, виявлення інформаційних потреб і збору вихідних даних для проєктування програмного забезпечення;
- оцінювати й вибирати методи та моделі розробки, впровадження, експлуатації програмних засобів та управління ними на всіх етапах життєвого циклу;
- розробляти й оцінювати стратегії проєктування програмних засобів, в тому числі з урахуванням впливу факторів різновекторного спрямування; обґрунтовувати, аналізувати й оцінювати прийняті проєктні рішення з точки зору якості кінцевого програмного продукту;
- аналізувати, оцінювати й вибирати методи, сучасні програмно-апаратні інструментальні та обчислювальні засоби, технології, алгоритмічні та програмні рішення для ефективного виконання конкретних виробничих завдань з програмної інженерії;

- обґрунтовано вибрати парадигми й мови програмування для вирішення прикладних завдань; застосовувати на практиці системні та спеціалізовані засоби, компонентні технології (платформи) та інтегровані середовища розробки програмного забезпечення;

- проводити аналітичне дослідження параметрів функціонування програмних систем для їхньої валідації та верифікації, а також проводити аналіз обраних методів, засобів автоматизованого проєктування та реалізації програмного забезпечення;

- знати й застосовувати сучасні професійні стандарти, а також інші нормативно-правові документи з інженерії програмного забезпечення;

- вміти приймати організаційно-управлінські рішення в умовах невизначеності;
- набувати нові наукові й професійні знання, вдосконалювати навички, прогнозувати розвиток програмних систем та інформаційних технологій;

- формулювати, експериментально підтверджувати, обґрунтовувати й застосовувати на практиці в процесі розробки програмного забезпечення конкурентоспроможні ідеї, методи, технології вирішення професійних, науково-технічних завдань в умовах невизначеності;

- здобувати необхідну інформацію з іншомовної літератури, аналізувати та вибирати необхідні для вирішення фахових наукових і прикладних завдань інформаційно-довідкові та науково-технічні ресурси й джерела знань з урахуванням сучасних досягнень науки й техніки, здійснювати різні види комунікації;

- організовувати командну роботу, управляти проєктами, підбирати команду проєкту, ефективно працювати в групі, визначати та розподіляти завдання з метою вирішення різноманітних дослідницьких та практичних завдань;

- проєктувати програмні засоби із використанням оптимальних сполучень багатокомпонентних стратегій.

На вивчення навчальної дисципліни відводиться 120 годин / 4 кредитів ECTS.

2. Інформаційний обсяг навчальної дисципліни

Змістовий модуль 1. Інструментальні засоби верифікації послідовних та паралельних програм алгоритмічної мовою.

Тема 1. Вступ до предмету. Огляд інструментальних засобів верифікації за допомогою формального доведення теорем стосовно властивостей програм. Статична та динамічна верифікація.

Тема 2. Система верифікації програм Splint. Системи VCC та Helix QAC для верифікації програм мовою C/C++ анотованих додатковими специфікаціями.

Тема 3. Засоби анотації функцій, інваріантів даних та циклів системи VCC. Формальне доведення відповідності програми її специфікаціям.

Змістовий модуль 2. Моделювання та верифікація паралельного програмного забезпечення сітьми Петрі (СП).

Тема 4. Подання основних типів процесів та ресурсів (змінних програми) СП.

Тема 5. Основні властивості моделі СП ідеальної програми. Поведінкові та структурні методи дослідження властивостей СП.

Тема 6. Нескінченні СП як засіб моделювання програмного забезпечення комп'ютерних мереж, ґридів та обчислювальних хмар. Засоби скінченної специфікації та основні методи властивостей у параметричній формі.

Тема 7. Модальні та темпоральні логіки як засіб специфікації властивостей програм. Мова LTL, μ -числення.

Змістовий модуль 3. Алгебра та числення процесів як засіб моделювання та верифікації розподілених програм.

Тема 8. Вступ до алгебри та числення процесів: дії, поведінка, еквівалентність та абстракції. Нотації Хоара та Беркстри. Основні типи даних.

Тема 9. Організація послідовних та паралельних процесів в алгебрі процесів. Лінеаризація послідовних та паралельних процесів, лінійне рівняння процесів та методи його розв'язання.

Тема 10. Модальні μ -числення як засіб специфікації запитів стосовно властивостей моделі. Логіка Хенесі-Мілнера, регулярні формули та модальності. Модальні формули з даними.

3. Рекомендована література

Базова

1. The VCC Manual Working draft, version 0.2, July 10, 2015, VCC User Manual (working draft, ver. 0.2).
2. A Practical Verification Methodology for Concurrent Programs, February 12, 2009, Technical Report MSR-TR-2009-15 Microsoft Research, Microsoft Corporation, One Microsoft Way Redmond, WA 98052 12p.
3. Э. М. Кларк, О. Грамберг, Д. Пелед, Верификация моделей программ: Model Checking. Пер. с англ. под ред. Р. Смелянского.- Издательство Московского центра непрерывного математического образования, Москва 2002, 416 с.
4. Zaitsev D.A. Clans of Petri Nets: Verification of protocols and performance evaluation of networks, LAP LAMBERT Academic Publishing, 2013, 292 p.
5. D. A. Zaitsev, I. D. Zaitsev and T. R. Shmeleva. Infinite Petri Nets: Part 1, Modeling Square Grid Structures. Complex Systems, 26 (2), 2017, pp.157-195.
6. J.F. Groote and M.R. Mousavi. Modeling and Analysis of Communicating Systems. The MIT Press, 2014.
7. Петрик М.Р. Моделювання програмного забезпечення : науково-методичний посібник / М.Р. Петрик, О.Ю. Петрик – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 200 с.
8. Г.В.Табунщик. Проектування та моделювання програмного забезпечення сучасних інформаційних систем Навч. посібник / Г.В.Табунщик, Т.І.Каплієнко, О.А.Петрова. – Запоріжжя, 2016. – 259 с.
9. Marieke Huisman, Vladimir Klebanov, Rosemary Monahan. VerifyThis 2012 A Program Verification Competition Int J Softw Tools Technol Transfer (2015) 17:647–657, DOI 10.1007/s10009-015-0396-8.

Додаткова

1. Berthomieu, B., Ribet, O.-P., Vernadat, F.: “The tool TINA-construction of abstract state space for Petri nets and time Petri nets,” *Int. J. Prod. Res.* 42(14), 2004, 2741–2756.
2. T.A.N. Engels, J.F. Groote, M.J. van Weerdenburg and T.A.C. Willemse. “Search algorithms for automated validation.” *Journal of Logic and Algebraic Programming* 78(4): pp.274-287, 2009.
3. Введение в формальные методы верификации программ: учебное пособие / А. С. Камкин. – Москва: МАКС Пресс, 2018. – 272 с.
4. Shmeleva T.R. Verification of network protocols using Finite Automata and Petri Nets. Methodical instructions for practical and laboratory works on course «Modeling and Optimization of Telecommunication Systems and Networks» / T.R.Shmeleva. – Odessa: A.S. Popov ONAT, 2015. – 32 p.
5. Д.А.Зайцев. Математичні моделі дискретних систем. Навчальний посібник з дисципліни .– Одеса, ОНАЗ, 2003, 35 с.

Інформаційні ресурси

1. MicrosoftVCC <https://rise4fun.com/vcc>. [Електронний ресурс]: [Веб-сайт].
2. mCRL2 <https://www.mcrl2.org/>. [Електронний ресурс]: [Веб-сайт]. Дата звернення: 07.09.2020
3. Tina <http://projects.laas.fr/tina>. [Електронний ресурс]: [Веб-сайт]. Дата звернення: 07.09.2020
4. Національна бібліотека України імені В. І. Вернадського [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Київ: НБУВ, 2013-2015. – Режим доступу: www.nbuv.gov.ua – Назва з екрана. Дата звернення: 07.09.2020
5. Електронний каталог Національної парламентської бібліотеки України [Електронний ресурс]: [політемат. база даних містить відом. про вітчизн. та зарубіж. кн., брош., що надходять у фонд НІБ України]. – Електронні дані (803 438 записів). – Київ: Нац. парлам. б-ка України, 2002-2015. – Режим доступу: catalogue.nplu.org . – Назва з екрана. Дата звернення: 07.09.2020
6. Український інститут інтелектуальної власності [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Київ: УІВ, 2017. – Режим доступу: <http://www.uivr.org> – Назва з екрана. Дата звернення: 07.09.2020

4. Форма підсумкового контролю успішності навчання

Формою контролю є іспит.

5. Засоби діагностики успішності навчання

Діагностика знань студентів здійснюється за допомогою:

- усного опитування;
- тестування;
- письмових поточних контрольних робіт;
- письмових екзаменаційних завдань.